



Universidad
Andrés Bello

Solución Actividad 1 Evaluada S_651

Estructuras de Datos

Docente: Pamela Landero Sepúlveda
p.landero@uandresbello.edu

Formar

Transformar



Universidad
Andrés Bello

Enunciado:

Para el ejercicio entregado como ejemplo de su docente: “Ingresar todas las notas parciales de los alumnos”, haga lo siguiente:

- 1) Crear una función que calcule el promedio por alumno (guardar en un arreglo).
- 2) Crear una función que muestre los alumnos reprobados (promedio menor que 4).
- 3) Crear una función que muestre el promedio general del curso.
- 4) Actualice el main de tal forma que haga la llamada de las funciones anteriores.

Así queda el main:

```
/** LIBRERIAS **/  
#include <stdio.h>  
#include <stdlib.h>  
  
/** PROTOTIPOS FUNCIONES DEL USUARIO **/  
float *calcularPromedio(float **pMatNotas, int pn, int pm);  
void mostrarReprobados(float *pArrProm, int pn);  
float calcularPromedioCurso(float *pArrProm, int pn);  
  
/** PROGRAMA PRINCIPAL **/  
int main(void) {  
    float **mtr_notasAlum; /*matriz*/  
    int i, j, n /*filas*/, m /*columnas*/;  
    printf ( " Filas : " );  
    scanf( " %d" , &n );  
    printf ( " Columnas : " );  
    scanf( " %d" , &m );  
  
    /* reserva memoria para las filas de la matriz:  
       cada fila representa a un alumno*/  
    mtr_notasAlum = (float **) malloc( n * sizeof ( float * ) );  
  
    for (i=0; i<n ; i ++ ) {  
        /*Por cada fila se reserva memoria para las columnas:  
           que corresponde a las notas de cada alumno*/  
        mtr_notasAlum[i] = (float *) malloc(m * sizeof ( float ) );  
    }  
}
```

```
/* Se ingresan las notas de cada alumno*/  
for (i=0; i<n ; i++) { //alumnos  
    for (j=0; j<m ; j++) { //Notas de cada alumno i  
        printf("Ingresa nota %d del alumno %d: ", j, i);  
        scanf("%f",&mtr_notasAlum[i][j] ); // (*(mat+i)+ j)  
    }  
    printf("\n");  
}  
  
/* Mostrar Notas Almacenadas */  
for (i=0; i<n ; i++) {  
    printf("NOTAS ALUMNO %d: ",i);  
    for (j=0; j<m ; j++){  
        printf(" %2.1f\t", mtr_notasAlum[i][j]);  
    }  
    printf("\n");  
}
```

/** AQUÍ SE LLAMAN LAS FUNCIONES **//

```
/* Liberación de memoria */  
for (i=0; i<n ; i ++){  
    free (mtr_notasAlum[i] );  
}  
free (mtr_notasAlum) ;  
return 0;  
} //***FIN MAIN***
```

/** COMIENZO PROGRAMACIÓN FUNCIONES **//

Llamada a Funciones desde el main

```
/** 1) LLAMA FUNCIÓN QUE CALCULA EL PROMEDIO POR ALUMNO **/  
float *arrProm = calcularPromedio(mtr_notasAlum, n, m); //Función retorna un arreglo  
for (i=0; i<n ; i ++){  
    ... printf(" \nPromedio ALUMNO %d: %2.1f", i, *(arrProm + i)); //Lo mismo arrProm[i]  
}
```

```
/** 2) LLAMA LA FUNCIÓN QUE MUESTRA ALUMNOS REPROBADOS **/  
mostrarReprobados(arrProm, n); //Función void
```

```
/** 3) LLAMA LA FUNCIÓN QUE CALCULA PROMEDIO DEL CURSO **/  
float promCurso = calcularPromedioCurso(arrProm, n); //Función retorna un float  
printf(" \n\nEL PROMEDIO DEL CURSO ES %2.1f: ", promCurso);
```

Función Calcular Promedio

```
/* 1) Función que calcula el promedio de notas por alumno  
y retorna el arreglo con las notas de cada alumno */  
float *calcularPromedio(float **pMatNotas, int pn, int pm){  
    float sumaNotas =0; //acumula la suma de las notas por cada alumno  
    int i,j;  
    /* Declarar e inicializar vector promedio */  
    float *prom = (float *) malloc(pn * sizeof ( float ) ) ;  
    for (i=0; i<pn ; i++) {  
        sumaNotas = 0;  
        for (j=0; j<pm ; j++){  
            sumaNotas = sumaNotas + pMatNotas[i][j];  
        }  
        /* Guarda el promedio del alumno i */  
        prom[i] = sumaNotas/pm;  
    }  
    return prom;  
}
```

Función Mostrar Alumnos Reprobados

```
/* 2) Función que muestra todos los alumnos reprobados */  
void mostrarReprobados(float *pArrProm, int pn){  
    int i;  
    for (i=0; i<pn ; i ++){  
        if(*(pArrProm + i)<4.0){  
            printf(" \nALUMNO %d REPROBO con NOTA: %2.1f", i, *(pArrProm + i)); //lo mismo pArrProm[i]  
        }  
    }  
}
```

Función Calcular Promedio del Curso

```
/* 3) Función que calcula el promedio de los alumnos  
   del curso */  
float calcularPromedioCurso(float *pArrProm, int pn){  
    int i;  
    float sumaNota=0;  
    for (i=0; i<pn ; i ++){  
        sumaNota = sumaNota + *(pArrProm + i);  
    }  
    return sumaNota/pn;  
}
```